Mining Goes Digital – Mueller et al. (Eds)

A two-stage solution approach for a shift scheduling problem with a simultaneous assignment of machines and workers

Cinna Seifi
Clausthal University of Technology, Clausthal-Zellerfeld, Germany

Marco Schulze
K+S Aktiengesellschaft, Kassel, Germany

Jürgen Zimmermann
Clausthal University of Technology, Clausthal-Zellerfeld, Germany

ABSTRACT: We consider a short-term production scheduling problem in a German potash underground mine where drill-and-blast mining operations have to be assigned to machines and workers and scheduled simultaneously. In addition, several mining-specific requirements have to be taken into account. In order to solve the problem at hand, we propose a two-stage solution approach. In the first stage, we apply a mixed-integer linear program where some time-consuming restrictions are neglected. Afterward, we modify the obtained schedule by integrating the necessary time intervals that were dismissed within the mathematical model. Since an existing heuristic solution procedure for the same problem is currently in use in a German potash mine, we will present results for computational experiments conducted on problem instances derived from real-world data in order to evaluate the performance of the two solution approaches.

1 PROBLEM DESCRIPTION AND LITERATURE REVIEW

This paper addresses a short-term production scheduling problem in a German potash underground mine that was already studied by Schulze & Zimmermann (2017) as well as Schulze et al. (2017) who proposed a rule-based constructive procedure. The extraction of the examined potash mine is done by room-and pillar mining method and the excavation of potash is based on drilling and blasting technique. This kind of underground mining is characterized by eight consecutive sub-steps, i.e., operations, that can be seen as a production cycle: scaling the roof, bolting the roof with expansion-shell bolts, drilling large diameter bore holes, removing the drilled material, drilling blast holes, filling the blast holes with an explosive substance, blasting, and transportation of the broken material to a crusher. For each operation, except blasting, one special mobile machine out of a set of identical or uniform machines is required that is handled by a worker with the corresponding qualification, i.e., skill. Hence, the underlying problem consists of the determination of a shift schedule where (i) a set of jobs[1] has to be selected and determined for execution, (ii) start times of the selected jobs have to be specified, and (iii) machines and workers have to be assigned to the jobs simultaneously while the individual skills of the workers as well as the technological-based precedence relations for the jobs have to be taken into account. The objective is the minimization of the average positive deviation between a predetermined quantity and the amount of extracted crude salt, cumulated over all operations.

Taking the aforementioned characteristics into consideration, we deal with two different problem types that have to be solved simultaneously, that is a machine scheduling problem on the one hand, and an employee timetabling problem on the other hand. The machine

1. Note that a job is characterized by the corresponding operation type from the production cycle and the place within the mine where the operation has to be executed.

scheduling problem can be classified as a variant of a so-called hybrid flow shop (HFS) scheduling problem, if we identify each mining operation from the production cycle as a stage, see Schulze et al. (2016). Note that we assume the mining operations as non-preemptable, because an assigned machine stays at the corresponding working place after an interruption (due to workers' breaks or end of a shift) and processing will be resumed at the next possible occasion. In comparison to the classical HFS scheduling problem, due to the short planning horizon of one single working shift, it is not possible to perform all mining operations at all working places. Thus, on the one hand, it has to be decided whether a job at a working place will be processed within the shift under consideration or not, and on the other hand, some possible interruption of the processing of a job at the end of the working shift must be taken into account.

The machines in the problem at hand are mobile and travel from job to job, therefore, no buffers are needed between the stages. Instead, we have to consider driving times between jobs that are processed by the same machine what results in sequence-dependent setup times. In addition, there are so-called technical services that have to be performed for the machines. Before processing the first operation assigned to a machine, a preventive maintenance (first technical services) must be done. After processing the last job assigned to a machine as well as before the end of the working shift if a job has to be interrupted, each machine must be cleaned and fueled (last technical services). Furthermore, for the case in which a worker changes his machine, he has to perform the first technical services for the new machine.

The second problem type, an employee timetabling problem, comprises the assignment of suitable workers to machines within a working shift. We assume that a worker can handle at most one machine at the same time and a machine can be operated by at most one worker at the same time. Since the workers have particular skills on different levels, not all workers can handle all machines. The different skill levels result in different handling times for each machine. Furthermore, the machines on each stage can have different speeds, which means that the processing time of a job depends on both the assigned worker and machine.

In order to generate a schedule that is accepted from the shift supervisor and that meets legal obligations, we furthermore have to satisfy the following mining-specific requirements.

R1: Due to legal regulations, a Δ-minute break for the workers has to be incorporated in the schedule within a predetermined time window. It has to be noted that the break can lead to a delay in the processing of a job.
R2: We have to consider disjunctive constraints for subsets of jobs that are physically close to each other (these jobs "belong" to so-called underground locations). Due to security reasons, it is not allowed that more than one machine is processing there at the same time, i.e., only one job in an underground location can be processed at any point in time.
R3: Although a consistent progress at all working places of the underground locations would be desirable, different excavation states appear, i.e., not the same operation can be performed there. In order to achieve a harmonized state, we prioritize the jobs in an underground location in a way that the progress strives for consistency.
R4: Jobs that are interrupted at the end of the pre-shift have a higher priority than the other jobs within the same underground location.
R5: The operations that symbolize the step drilling large diameter bore holes are non-interruptible. If those operations cannot be finished until the end of the corresponding working shift, they must not be started.
R6: The number of assigned worker to a machine is limited by two. That means no more than two workers can be assigned to a machine and a worker is not allowed to work on more than two machines during a working shift.

In the literature, most works concerning machine scheduling neglect the assignment of workers, while in the field of employee timetabling (i.e., staff scheduling and personnel assignment) the machine scheduling problem is rarely considered. Therefore, we confined our literature review to the integrated employee timetabling and machine scheduling problems. An overview of the studied literature is given in Table 1, where the abbreviations in column *production environment* characterize, whether the authors analyzed a flow shop (FSP), job

Table 1. Literature review.

	Production environment	Objective function	Formulation
Daniels & Mazzola (1994)	FSP	makespan	time-indexed
Daniels et al. (2004)	FSP	makespan	time-indexed
Huq et al. (2004)	FSP	multi-obj.	seq.-based
Artigues et al. (2006)	JSP	empl. cost	time-indexed
Artigues et al. (2009)	JSP	multi-obj.	seq.-based
Puttkammer et al. (2011)	FSP	multi-obj.	time-indexed
Mencia et al. (2013)	JSP	flow time	
Ramya & Chandrasekaran (2014)	JSP	empl. cost	time-indexed
Frihat et al. (2014)	HJSP	empl. cost	seq.-based
Benavides et al. (2014)	FSP	makespan	seq.-based
Guyon et al. (2014)	JSP	empl. cost	
Agnetis et al. (2014)	JSP	makespan	seq.-based
Campos Ciro et al. (2016)	OSP	flow time	seq.-based
Ahmadi-Javid & Hooshangi-Tabrizi (2017)	JSP	makespan	seq.-based
Santos et al. (2018)	JSP	throughput	time-indexed

shop (JSP), open shop (OSP), or hybrid job shop (HJSP) scheduling problem. Moreover, we indicate what kind of *objective function* as well as *formulation* is considered within the corresponding study.

In brief, our literature review shows that the approaches discussed in the studies are not suitable for the problem at hand what is mainly due to the aspects that we consider (i) a selection of jobs, (ii) setup times for machines and workers, (iii) possible interruption of jobs at the end of the shift, (iv) breaks that could delay the processing of jobs, and (v) that the workers can change their machine within a working shift. In the next section, we introduce a two-stage approach to tackle the problem.

2 TWO-STAGE APPROACH

In our two-stage approach, we first solve a relaxation of the problem described in the previous section using a MIP solver and then, we repair the solution found and generate a feasible one. First, we describe the relaxation (**R-Model**) of our short-term scheduling problem, where some restrictions concerning the breaks or technical services are omitted.

Let J be the set of jobs in the underground mine under consideration. Binary decision variables b_j are 1 if $j \in J$ is processed. Moreover, we introduce binary decision variables x_{jw} and y_{jm} that are 1 if j is processed by worker $w \in W_j$ and machine $m \in G_j$, respectively. G_j and W_j are subsets of the set of available machines G and workers W that can process job j. We also define binary decision variables z_{jwm} that are 1 if worker w and machine m are assigned to job j.

$$\sum_{m \in G_j} y_{jm} = b_j \quad \forall j \in J \tag{1}$$

$$\sum_{w \in W_j} x_{jw} = b_j \quad \forall j \in J \tag{2}$$

$$x_{jw} + y_{jm} \leq 1 + z_{jwm} \quad \forall j \in J, \forall w \in W_j, \forall m \in G_j \tag{3}$$

$$z_{jwm} \leq x_{jw} \quad \forall j \in J, \forall w \in W_j, \forall m \in G_j \tag{4}$$

$$z_{jwm} \leq y_{jm} \quad \forall j \in J, \forall w \in W_j, \forall m \in G_j \tag{5}$$

Let PT_{jwm} be the given parameter that denotes the processing time of job j by worker w and machine m. The actual processing time of job j is then $p_j = \sum_{w \in W_j} \sum_{m \in G_j} z_{jwm} \cdot PT_{jwm}$. In this paper, we use a sequence-based formulation for our **R-Model**. So, we introduce binary decision variables v_{jr} that are 1 if job j is completed before job r is started.

$$v_{jr} + v_{rj} \leq 1 \quad \forall j, r \in J : j \neq r \tag{6}$$

We have to consider a break for each worker (see **R1**) that leads to the absence of this worker in a specific time. Each worker w has to make a Δ-minute break so that the start time of the break ρ_w lies in a predefined interval $[varphi^{\alpha}, \varphi^{\omega}]$. In our relaxation, we do not allow that the processing of a job overlaps the break of the worker who processes this job. However, the break may overlap the drive between two jobs or the technical services that may be executed for a machine.

$$\varphi^{\alpha} \leq \rho_w \quad \forall w \in W \tag{7}$$

$$\rho_w \leq \varphi^{\omega} \quad \forall w \in W \tag{8}$$

$$S_j \leq \rho_w + \left(1 - \omega_j^s\right) M + \left(1 - x_{jw}\right) M \quad \forall j \in J, \forall w \in W_j \tag{9}$$

$$\rho_w + \boxtimes \leq S_j + \omega_j^s M + \left(1 - x_{jw}\right) M \quad \forall j \in J, \forall w \in W_j \tag{10}$$

$$\rho_w \leq S_j + p_j + \left(1 - \omega_j^e\right) M + \left(1 - x_{jw}\right) M \quad \forall j \in J, \forall w \in W_j \tag{11}$$

$$S_j + p_j \leq \rho_w + \omega_j^e M + \left(1 - x_{jw}\right) M \quad \forall j \in J, \forall w \in W_j \tag{12}$$

$$\omega_j^s + \omega_j^e \leq 1 + \delta_j \quad \forall j \in J \tag{13}$$

$$\delta_j \leq \omega_j^s \quad \forall j \in J \tag{14}$$

$$\delta_j \leq \omega_j^e \quad \forall j \in J \tag{15}$$

If the start time of the break of a worker, who processes job j, is during the processing of j ($\omega_j^s = 1$ and $\omega_j^e = 1$), binary decision variable δ_j takes the value of 1 and a Δ-minute break must be considered for the duration of j, additionally.

At the beginning of the working shift, the first technical services td_m^{α} must be performed on machine m and the assigned worker drives d_{0jm} time units to the first job.

$$\sum_{m \in G_j} (td_m^{\alpha} + d_{0jm}) \cdot y_{jm} \leq S_j + \left(1 - b_j\right) M \quad \forall j \in J \tag{16}$$

At each working place in an underground location, several operation types must be executed in a specific order related to the prescribed production cycle. Let ul_j be the underground location of j, ml_j be the working place of j in ul_j, and $order_j$ be the position of j in the given order for ml_j. In a working place, always job j with the minimum value of $order_j$ must be completed before any job r with a greater value of $order_r$ can be started.

$$b_r \leq b_j \quad \forall j, r \in J : j \neq r, ul_j = ul_r, ml_j = ml_r, order_j < order_r \tag{17}$$

$$\begin{gathered} S_j + p_j + \delta_j \cdot \Delta \leq S_r + \left(2 - b_j - b_r\right) M \\ \forall j, r \in J : j \neq r, ul_j = ul_r, ml_j = ml_r, order_j < order_r \end{gathered} \tag{18}$$

For jobs that are processed by the same worker, a precedence relation must be considered. As mentioned in Sect. 1, if a worker changes his machine, he has to go to the new machine (transfer time), has to do the first technical services, and he can drive the machine to the location of the new job. In our relaxed model, we neglect the time for the case, where a worker changes his machine.

$$S_j + p_j + \delta_j \cdot \Delta \leq S_r + \left(2 - x_{jw} - x_{rw}\right)M + \left(1 - v_{jr}\right)M$$
$$\forall j,r \in J : j \neq r, \forall w \in W_j \cap W_r \tag{19}$$

$$S_r + p_j + \delta_r \cdot \Delta \leq S_j + \left(2 - x_{jw} - x_{rw}\right)M + v_{jr}M$$
$$\forall j,r \in J : j \neq r, \forall w \in W_j \cap W_r \tag{20}$$

Moreover, if two jobs are processed by the same machine, a driving time between the jobs must be taken into account.

$$S_j + p_j + d_{jrm} + \delta_j \cdot \Delta \leq S_r + \left(2 - y_{jm} - y_{rm}\right)M + \left(1 - v_{jr}\right)M$$
$$\forall j,r \in J : j \neq r, \forall m \in G_j \cap G_r \tag{21}$$

$$S_r + p_j + d_{rjm} + \delta_r \cdot \Delta \leq S_j + \left(2 - y_{jm} - y_{rm}\right)M + v_{jr}M$$
$$\forall j,r \in J : j \neq r, \forall m \in G_j \cap G_r \tag{22}$$

After processing the last job on a machine, last technical services must be performed for the machine. Let *Shift* be the duration of the working shift. The following constraints guarantee that if the processing of a job exceeds *Shift* ($id_j = 1$), continuous decision variable $grad_{jwm}$ specifies, which percentage of *j* is achieved during the shift.

$$S_j + p_j + \delta_j \cdot \Delta + \sum_{m \in G_j} td_m^{\omega} \cdot y_{jm} \leq Shift + id_j M \quad \forall j \in J \tag{23}$$

$$Shift \leq S_j + p_j + \delta_j \cdot \Delta + \sum_{m \in G_j} td_m^{\omega} \cdot y_{jm} + \left(1 - id_j\right)M \quad \forall j \in J \tag{24}$$

$$S_j + \delta_j \cdot \Delta + \sum_{w \in W_j} \sum_{m \in G_j} grad_{jwm} \cdot PT_{jwm} + \sum_{m \in G_j} td_m^{\omega} \cdot y_{jm} \leq Shift + \left(1 - id_j\right)M \, \forall j \in J \tag{25}$$

$$S_j + \delta_j \cdot \Delta + \sum_{w \in W_j} \sum_{m \in G_j} grad_{jwm} \cdot PT_{jwm} + \sum_{m \in G_j} td_m^{\omega} \cdot y_{jm} \geq Shift \cdot id_j \, \forall j \in J \tag{26}$$

Consequently, if a job is processed ($b_j = 1$) and its duration does not exceed the working shift ($id_j = 0$), $grad_{jwm}$ must take the value of 1.

$$\sum_{w \in W_j} \sum_{m \in G_j} grad_{jwm} \leq \left(id_j + b_j\right)M \quad \forall j \in J \tag{27}$$

$$1 - \left(id_j - b_j + 1\right)M \leq \sum_{w \in W_j} \sum_{m \in G_j} grad_{jwm} \quad \forall j \in J \tag{28}$$

For the other mining-specific requirements **R2**–**R5**, we formulate the following constraints. Note that we write $j \prec r$ if job *j* must be completed before job *r* can be started.

R2:

$$S_j + p_j + \delta_j \cdot \Delta \leq S_r + \left(2 - b_j - b_r\right)M + \left(1 - v_{jr}\right)M$$
$$\forall j,r \in J : j \neq r, ul_j = ul_r \tag{29}$$

$$S_r + p_j + \delta_r \cdot \Delta \leq S_j + \left(2 - b_j - b_r\right)M + v_{jr}M$$
$$\forall j,r \in J : j \neq r, ul_j = ul_r \tag{30}$$

R3:

$$b_r \le b_j \quad \forall j,r \in J : j \ne r, ul_j = ul_r, j \prec r \tag{31}$$

$$S_j + p_j + \delta_j \cdot \Delta \le S_r + \left(2 - b_j - b_r\right) M \quad \forall j,r \in J : j \ne r, ul_j = ul_r, j \prec r \tag{32}$$

R4:

$$b_r \le b_j \quad \forall j,r \in J : j \ne r, ul_j = ul_r, started_j = 1, started_r = 0 \tag{33}$$

R5:

$$id_j = 0 \quad \forall j \in J : type_j = 4 \tag{34}$$

To realize **R6**, we introduce binary decision variables ma_{wm} that are 1 if w is assigned to m.

$$y_{jm} + x_{jw} \le 1 + ma_{wm} \quad \forall j \in J, \forall w \in W_j, \forall m \in G_j \tag{35}$$

$$\sum_{m \in G} ma_{wm} \le 2 \quad \forall w \in W \tag{36}$$

$$\sum_{w \in W} ma_{wm} \le 2 \quad \forall m \in G \tag{37}$$

Let ton_j be the expected amount of material after processing of job j and ton_k^{pre} be the predetermined quantity (target value) for production step k. We can determine the lower deviation from ton_k^{pre} for each production step by the following constraints.

$$ton_k^{pre} - \sum_{j \in J : type_j = k} \sum_{w \in W_j} \sum_{m \in G_j} grad_{jwm} \cdot ton_j \le dev_k \quad \forall k \in K \tag{38}$$

Our goal is to have a consistent progress so that the following function must be minimized.

$$\sum_{k \in K} dev_k^2$$

If we determine the maximum lower deviation as follows:

$$dev_k \le dev^{max} \quad \forall k \in K, \tag{39}$$

we can then approximate the quadratic objective function by the following linear one.

$$\sum_{k \in K} dev_k + dev^{max} \tag{40}$$

After finding a solution for **R-Model** (Min. (40) s.t. (1)–(39)), the solution is used as an input for Algorithm 1 to generate a feasible solution for the problem instance at hand. In Algorithm 1, we first determine the sequence of the processing of jobs for each machine and each worker. After that, for each machine m, between two consecutive jobs j and r that are processed by different workers, a first technical service must be inserted before starting r. Subsequently, start times of all of the jobs that have to be started after the completion of r must be updated. For each worker w, if w changes his machine and goes from machine m to m', we eventually have to consider last technical services if w processed the last task on m. The worker then goes to the parking location of m', performs first technical services, and drives m' to the location of the next job. Consequently, start times of all of the related jobs must be updated. Then, we check if there are overlaps between breaks of workers and the activities that workers have to perform. In this case, we consider the effect of workers' breaks on start times or durations of jobs and update the start times of all of the affected jobs. The steps above are repeated until there are no more changes in start times of jobs.

Algorithm 1. Repair solution.

Input: problem instance D, solution of **R-Model**
repeat
  For each machine, determine the sequence of processed jobs by this machine;
  For each worker, determine the sequence of processed jobs by this worker;
  for all processed jobs j **do**
  $S_j^1 = S_j$
  for all machines m **do**
    if two consecutive jobs j and r on m are performed by different workers **then**
      Insert the time for first technical services for m before S_r
      for all jobs j' with $v_{rj'} = 1$ **do**
        update $S_{j'}$
  for all workers w **do**
    if two consecutive jobs j and r on w are performed by different machines **then**
      Insert the potential last technical services for the machine assigned to j, the transfer time, first technical services for the machine assigned to r, and the driving time from the direct predecessor of r on the assigned machine to r
      for all jobs j' with $v_{rj'} = 1$ **do**
        update $S_{j'}$
  for all workers w **do**
    if the break of w, who processes job j, overlaps any of the first technical services for the machine assigned to j, driving times to j, the processing of j, or the last technical services for the machine assigned to j **then**
      Consider the break of w for S_j or the duration of j
      for all jobs j' with $v_{jj'} = 1$ **do**
        update $S_{j'}$
  for all jobs j with $type_j = 4$ **do**
    Eliminate j if the processing of j exceeds the duration of the working shift
until $S_j^1 = S_j \forall_j$
Determine the objective value according to the new schedule
return The feasible schedule

3 COMPUTATIONAL STUDY

To show the suitability of our proposed solution approach, we compare the results of our two-stage approach with the heuristic procedure introduced by Schulze & Zimmermann (2017). For this purpose, we generated 100 test instances based on the case study presented in Schulze & Zimmermann (2017), which depict realistic problems in a German potash underground mine. In Algorithm 2, an overview of the constructive heuristic approach is given (for more details see Schulze & Zimmermann (2017) and Schulze (2016)).

Algorithm 2. Constructive heuristic introduced by Schulze & Zimmermann (2017).

Initialization (* *construction* *)
Priority-based scheduling
Staff changes (* *improvement* *)
repeat
  Scheduling downstream operations
  Staff changes
  Replenishment
until total amount of potash cannot be increased
Job reassignment
Insertion of **technical services** (* *post-processing* *)
Insertion of **breaks** for workers
return solution

Table 2. Comparison of the approaches.

	Number of best solutions found	Gap to the best solution found
Two-Stage approach	70	6.1% (20.3%)
Constructive heuristic	30	45.2% (64.6%)

The heuristic procedure is embedded in a multi-start algorithm, where jobs, machines, and workers are chosen based on selection probabilities that are determined by priority values. For the heuristic approach in this paper, we use the setting that is currently used in the underground mine under consideration. In the corresponding assigning method, jobs and workers are randomly chosen, and machines are selected regarding the shortest driving times to the selected job or regarding the shortest processing time based on the selected job and the selected worker.

All tests are executed on an Intel i7–7700 K@4.20 GHz machine with 64 GB RAM under Windows 10. The heuristic algorithm is implemented in Xpress IVE 8.4. For the two-stage approach, we used GAMS 25.1 and GUROBI solver 8.1.0 to solve **R-Model** and C++ to generate a feasible solution with the aid of Algorithm 1. Since we schedule only one working shift, we set an upper time limit of 900 seconds for both approaches that symbolizes a typical duration of a shift handover.

To compare the results achieved by the procedures, we use the value of $\sum_{k \in K} dev_k^2$ that shows how consistent the desired progress could be implemented at the end of the working shift compared to the given state at the beginning of the working shift.

Table 2 presents the number of best solutions found and an average gap to the best solution found. Let S_i^* be the solution found for instance i by procedure * and S^{best} be the best solution found. We calculate $\frac{S_i^* - S^{best}}{S_i^*}$ to determine the gap for instance i (gap_i). The numbers presented under "Gap to the best solution found" are obtained by arithmetic averaging over all instances. Note that the number in parentheses is the obtained gap by arithmetic averaging over the number of the instances for which the solution found is not equal to the best solution found (i.e., $\text{gap}_i \neq 0$).

We see that the two-stage approach can find for 70 instances the best solution, where the solutions found for the other 30 instances are 20.3% far from the best solution found. On the other hand, the solutions found by the constructive heuristic are on average 45.2% worse than the best solutions found. This number gets significantly worse (64.6%) if we make an average over the 70 instances for which the heuristic could not find the best solution. So, we can conclude that our two-stage approach performs quite promising.

4 CONCLUSION

In this paper, we consider a shift scheduling problem where machines and workers are simultaneously assigned to a selection of the available jobs. We formulate a relaxation of the problem described in Sect. 1 and introduced an algorithm to generate feasible solutions using the solution achieved by the relaxation. The results of a preliminary performance analysis using realistic instances show that the solutions of our proposed two-stage approach clearly outperform the solutions which are currently generated by a constructive heuristic procedure.

Future work concerns the development of good lower bounds for our problem. Moreover, the total output can additionally be taken into account. Considering the trade-off between the presented objective function and the total excavated amount of material could provide new insights.

REFERENCES

Agnetis, A., Murgia, G., & Sbrilli, S., 2014. A job shop scheduling problem with human operators in handicraft production. *International Journal of Production Research* 52(13): 3820–3831.

Ahmadi-Javid, A. & Hooshangi-Tabrizi, P., 2017. Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment: A mathematical formulation and an anarchic society optimization algorithm. *Computers & Operations Research* 84: 73–91.

Artigues, C., Gendreau, M. & Rousseau, L.M., 2006. A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 67–84). Berlin, Heidelberg: Springer.

Artigues, C., Gendreau, M., Rousseau, L.M. & Vergnaud, A., 2009. Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Computers & Operations Research* 36(8): 2330–2340.

Benavides, A.J., Ritt, M. & Miralles, C., 2014. Flow shop scheduling with heterogeneous workers. *European Journal of Operational Research* 237(2): 713–720.

Campos Ciro, G., Dugardin, F., Yalaoui, F. & Kelly, R., 2016. Open shop scheduling problem with a multi-skills resource constraint: A genetic algorithm and an ant colony optimisation approach. *International Journal of Production Research* 54(16): 4854–4881.

Daniels, R.L. & Mazzola, J.B., 1994. Flow shop scheduling with resource flexibility. *Operations Research* 42(3): 504–522.

Daniels, R.L., Mazzola, J.B. & Shi, D., 2004. Flow shop scheduling with partial resource flexibility. *Management Science* 50(5): 658–669.

Frihat, M., Sadfi, C. & Hadj-Alouane, A.B., 2014. Optimization of integrated employee timetabling and hybrid job shop scheduling under time lag constraints. In *International Conference on Control, Decision and Information Technologies (CoDIT)*, (pp. 282–287). IEEE.

Guyon, O., Lemaire, P., Pinson, E. & Rivreau, D., 2014. Solving an integrated job-shop problem with human resource constraints. *Annals of Operations Research* 213(1): 147–171.

Huq, F., Cutright, K. & Martin, C., 2004. Employee scheduling and makespan minimization in a flow shop with multi-processor work stations: a case study. *Omega* 32(2): 121–129.

Mencia, C., Sierra, M.R. & Varela, R., 2013. An efficient hybrid search algorithm for job shop scheduling with operators. *International Journal of Production Research* 51(17): 5221–5237.

Puttkammer, K., Kleber, R., Schulz, T. & Inderfurth, K., 2011. Simultane Maschinenbelegungs- und Personaleinsatzplanung in KMUs anhand eines Fallbeispiels aus der Druckereibranche. In Sucky, E., Asdecker, B., Dobhan, A., Haas, S. & Wiese, J. (eds), *Logistikmanagement: Herausforderungen, Chancen und LÃ¶sungen*: 229–247. University of Bamberg Press.

Ramya, G. & Chandrasekaran, M., 2014. Shuffled frog leaping algorithm approach to employee timetabling and job shop scheduling. *International Journal of Internet Manufacturing and Services 4 & 25* 3(3): 178–203.

Santos, F., Fukasawa, R. & Ricardez-Sandoval, L., 2018. An integrated personnel allocation and machine scheduling problem for industrial size multipurpose plants. *IFAC-PapersOnLine* 51(18): 156–161.

Schulze, M., 2016. Ein hierarchischer Ansatz zur LÃ¶sung von Ablaufplanungsproblemen im Bergbau: Darstellung am Beispiel des Ã–rterbaus. Aachen: Shaker.

Schulze, M., Mathiak, T. & Haney, C., 2017. Using operations research techniques to increase efficiency in German potash mining. In *Application of Computers and Operations Research in the Mineral Industry; Proc. intern. symp., Golden, Colorado, 2017* 5: 9–15.

Schulze, M., Rieck, J., Seifi & C., Zimmermann, J., 2016. Machine scheduling in underground mining: An application in the potash industry. *OR Spectrum* 38(2): 365–403.

Schulze, M., Zimmermann, J., 2017. Staff and machine shift scheduling in a German potash mine. *Journal of Scheduling* 20(6): 635–656.